

日 本 国 特 許 庁

JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2003年 3月17日

出 願 番 号

Application Number:

特願2003-072552

[ ST.10/C ]:

[ JP 2003-072552 ]

出 願 人

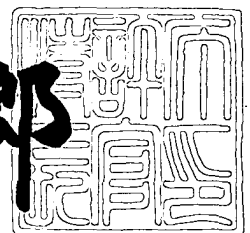
Applicant(s):

任天堂株式会社

2003年 6月26日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

太田信一郎



出証番号 出証特2003-3050787

【書類名】 特許願

【整理番号】 ND-0123P

【あて先】 特許庁長官殿

【国際特許分類】 A63F 13/00

【発明者】

【住所又は居所】 京都府京都市南区上鳥羽鉾立町 1 1 番地 1 任天堂株式会社内

【氏名】 太田 敬三

【特許出願人】

【識別番号】 000233778

【氏名又は名称】 任天堂株式会社

【代理人】

【識別番号】 100098291

【弁理士】

【氏名又は名称】 小笠原 史朗

【手数料の表示】

【予納台帳番号】 035367

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9201609

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 シャドウボリューム生成プログラム及びゲーム装置

【特許請求の範囲】

【請求項 1】 3 次元の仮想空間に配置されたオブジェクトが落とす影を描画するために利用されるシャドウボリュームをコンピュータに生成させるためのシャドウボリューム生成プログラムであって、前記コンピュータに、

仮想空間に設置された光源を視点として、少なくとも前記影を落とすオブジェクトを含む所定領域内の各ピクセルに対応する Z 値を Z バッファに書き込むステップと、

前記シャドウボリュームの元となる平面状オブジェクトを構成する複数のポリゴンの各頂点の厚み方向に関する位置を、前記 Z バッファに書き込まれた各ピクセルの Z 値に応じて決定することによって、該平面状オブジェクトから前記シャドウボリュームを生成するステップとを実行させる、シャドウボリューム生成プログラム。

【請求項 2】 前記平面状オブジェクトの形状は、X 座標および Z 座標の組み合わせが異なる複数の頂点によって規定されており、

前記シャドウボリューム生成ステップでは、前記平面状オブジェクトの各頂点の Y 座標が、前記 Z バッファに書き込まれた各ピクセルの Z 値に応じて決定されることを特徴とする、請求項 1 に記載のシャドウボリューム生成プログラム。

【請求項 3】 前記光源が点光源であり、

前記シャドウボリューム生成ステップは、前記平面状オブジェクトの各頂点の面方向に関する位置を、前記 Z バッファに書き込まれた各ピクセルの Z 値に応じて決定するステップを含むことを特徴とする、請求項 1 に記載のシャドウボリューム生成プログラム。

【請求項 4】 前記平面状オブジェクトの形状は、X 座標および Z 座標の組み合わせが異なる複数の頂点によって規定されており、

前記シャドウボリューム生成ステップでは、前記平面状オブジェクトの各頂点の X 座標および Z 座標が、前記 Z バッファに書き込まれた各ピクセルの Z 値に応じて決定されることを特徴とする、請求項 3 に記載のシャドウボリューム生成プ

プログラム。

【請求項５】 前記シャドウボリューム生成ステップで生成されたシャドウボリュームを、該シャドウボリュームの厚み方向が前記光源の光線方向と一致するように前記仮想空間に仮想的に配置するステップと、

仮想的に配置したシャドウボリュームを用いて前記影を落とすオブジェクトの影を描画するステップとをさらに前記コンピュータに実行させることを特徴とする、請求項１に記載のシャドウボリューム生成プログラム。

【請求項６】 ３次元の仮想空間に配置されたオブジェクトが落とす影を描画するために利用されるシャドウボリュームを生成するゲーム装置であって、

Zバッファと、

仮想空間に設置された光源を視点として、少なくとも前記影を落とすオブジェクトを含む所定領域内の各ピクセルのZ値を前記Zバッファに書き込むZ値書込手段と、

前記シャドウボリュームの元となる複数のポリゴンからなる平面状オブジェクトの各頂点の厚み方向に関する位置を、前記Z値書込手段によって前記Zバッファに書き込まれた各ピクセルのZ値に応じて決定することによって、該平面状オブジェクトから前記シャドウボリュームを生成するシャドウボリューム生成手段とを備えるゲーム装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明はシャドウボリューム生成プログラムに関し、より特定的には、シャドウボリューム技法において利用されるシャドウボリュームを生成するためのシャドウボリューム生成プログラムに関する。

【０００２】

【従来の技術】

シャドウボリューム技法とは、主に影の表現に使われる技法のことであって、影を落とすモデルによって光が遮られる空間（一般にシャドウボリュームと呼ばれる。）に対応した空間モデル（この空間モデル自体がシャドウボリュームと呼

ばれることもある。)を用意し、この空間モデルを特殊な方法でレンダリングすることによって(この空間モデル自身は描かれない)、影が落ちるモデルの、空間モデルの内部に位置する部分の色だけを暗くする方法である(例えば、特許文献1参照。)。例えば影を落とす物体が球体であれば、その物体によって光が遮られる空間は円柱状となる。

#### 【0003】

以下の説明では、影を落とすモデルを「影オブジェクト」と呼び、影の形を決定する空間モデルを「シャドウボリューム」と呼ぶことにする。

#### 【0004】

このシャドウボリューム技法で問題となるのが、シャドウボリュームの形をどのように決めるかである。シャドウボリュームの形は、影オブジェクトの形状アニメーションや、光源方向の変化によって様々に変化する。

#### 【0005】

シャドウボリュームの形を決定する従来の方法の一例として、影オブジェクトを光源方向から見たときに表を向いているポリゴンと裏を向いているポリゴンに共有されているエッジを輪郭線としてシャドウボリュームの形を決定するものがある。

#### 【0006】

例えば、図15(a)に示すように斜めに傾いた直方体(影オブジェクト)に光源からの光が当たっている場合、影オブジェクトを光源から見ると図15(b)のようになっている。従来の方法では、この影オブジェクトにおいて、光源に対して表向きのポリゴン(図15(c))と、光源に対して裏向きのポリゴン(図15(d))をそれぞれ検出し、光源に対して表向きのポリゴンと、光源に対して裏向きのポリゴンによって共有されるエッジを抽出する(図15(e)の実線)。このエッジがシャドウボリュームの断面の輪郭線となる。つまり、図15(e)の実線で示した形を断面に持つ筒状のモデルがシャドウボリュームとして使用される。

#### 【0007】

【特許文献1】

【0008】

【発明が解決しようとする課題】

上記のような従来の方法では、まず、各エッジを共有する面を示すデータと、全ての面の法線情報が必要となる。これらは通常のグラフィックデータには含まれず、特別なデータコンバータを作成した上で、メモリ上に置いておくことになる。

【0009】

そして、実際の処理では様々な例外処理を必要とするので、専用のプロセッサを実現できず、CPUによる時間をかけた処理が必要となる。

【0010】

具体的には、全てのエッジについて、そのエッジに接している面の法線とライト方向との内積を計算し、ライト方向に対して表を向いた面と裏を向いた面に共有されるエッジが選出される。そして、選出されたエッジのうち、同一の頂点でつながっているものが順次選出され、最終的に輪となる1つながりの線状のデータが作成される。これがシャドウボリユームの断面のデータとなる。そして、この断面をライト方向に伸ばして筒状にしたものがシャドウボリユームの形状データとなる。

【0011】

しかしながら、このようにしてシャドウボリユームが難なく作成されるのはまれである。例えば、図16(a)に示すように、直方体に真上から光が当たる場合のように影オブジェクトが光源方向に対して平行な面を有する場合には、図16(b)に示すように、輪郭線として複数のエッジが選択可能となり、エッジの選択が困難となる。また、図16(c)に示すように、雪ダルマのような凹みのある物体に斜めから光が当たる場合には、図16(d)に示すように、輪郭線が途切れてしまうため、特別な処理が必要となる。また、図16(e)に示すように、雪ダルマのような凹みのある物体に真上から光が当たる場合には、複数の輪郭線が検出され、その結果、シャドウボリユームが複数発生して処理の手続きが倍になる。

#### 【0012】

一般に、上記のような異常な状態は頻繁に発生し、これに対処するためにはたくさんの例外処理プログラムを用意したり、元々のデータに様々な細工をしたりする必要がある。しかしながら、このような作業はとても困難であり、処理負担も大きい。特にゲーム機においてゲーム画像を生成する場合には、1フレーム（1/30秒または1/60秒）内にゲーム画像を生成する必要があり、このように負担の大きい処理を採用することはできない。そこで普通は、そのような異常な状態が絶対に起こり得ないような単純な形状データをシャドウボリューム作成用に用意し、これを利用してシャドウボリュームが作成される。しかしながらこのような方法で影を描画した場合、本来の影とは異なり、かなり省略された影しか表現できない。

#### 【0013】

それゆえに本発明は、例外的な処理を必要とせず、常に一定の簡単な処理でシャドウボリュームを作成することができるシャドウボリューム生成プログラムを提供することを目的とする。

#### 【0014】

##### 【課題を解決するための手段】

上記の課題を解決するために本発明は以下の構成を採用した。ただし、括弧内の参照符号等は、本発明の理解を助けるために後述する実施形態との対応関係を示したものであって、本発明の範囲を何ら限定するものではない。

#### 【0015】

本発明のシャドウボリューム生成プログラムは、3次元の仮想空間に配置されたオブジェクト（2）が落とす影を描画するために利用されるシャドウボリューム（4）をコンピュータ（10、11）に生成させるためのプログラムである。このプログラムは、コンピュータ（10、11）に、まず、仮想空間に設置された光源を視点として、少なくとも影を落とすオブジェクト（2）を含む所定領域内の各ピクセルに対応するZ値をZバッファ（15）に書き込むステップ（S201）を実行させ、続いて、シャドウボリュームの元となる平面状オブジェクト（3）を構成する複数のポリゴンの各頂点の厚み方向（Y軸方向）に関する位置

(p y) を、Zバッファ(15)に書き込まれた各ピクセルのZ値(Z value)に応じて決定する(S206)ことによって、この平面状オブジェクト(3)からシャドウボリューム(4)を生成するステップを実行させる。

#### 【0016】

本発明のゲーム装置は、3次元の仮想空間に配置されたオブジェクト(2)が落とす影を描画するために利用されるシャドウボリューム(4)を生成するものであって、Zバッファ(15)と、Z値書込手段(ステップS201を実行するCPU10ないしGPU11)と、シャドウボリューム生成手段(ステップS204を実行するCPU10ないしGPU11)とを備える。Z値書込手段は、仮想空間に設置された光源を視点として、少なくとも影を落とすオブジェクト(2)を含む所定領域内の各ピクセルのZ値をZバッファ(15)に書き込む。シャドウボリューム生成手段は、シャドウボリュームの元となる平面状オブジェクト(3)を構成する複数のポリゴンの各頂点の厚み方向(Y軸方向)に関する位置(p y)を、Z値書込手段によってZバッファ(15)に書き込まれた各ピクセルのZ値(Z value)に応じて決定することによって、この平面状オブジェクト(3)からシャドウボリューム(4)を生成する。

#### 【0017】

#### 【発明の実施の形態】

以下、本発明の一実施形態について説明する。ただし、実施形態について詳細に説明する前に、まず、本発明の原理について簡単に説明する。

#### 【0018】

本発明は、あたかも非常に柔らかい1枚の布を影オブジェクトに対して光源方向から被せたときの布の形状を、シャドウボリュームの形状として利用するという発想に基づいている。

#### 【0019】

例えば、図1(a)に示すように、ひょうたん形の影オブジェクトに対して上方から光が当たっているときに、この影オブジェクトに対して光源の方向から柔らかな布を被せると(図1(b)、図1(c))、最終的に図1(d)の状態になる。図1(d)において布によって囲まれた空間は、ひょうたん形の影オブジ



ェクトのシャドウボリュームと一致するため、この布の形状をシャドウボリュームの形状として利用することによって、正確な形状のシャドウボリュームが得られることになる。他の例として、ドーナツ型の影オブジェクトに対して光が当たっているときの様子を図1（e）～図1（h）に示す。このようにして生成されたシャドウボリュームを用いてシャドウボリューム技法による影の描画を行うと、図2（a）および図2（b）に実線で示すような影を表示することができる。

#### 【0020】

以下、本実施形態の一実施形態に係るゲームシステムについて説明する。

図3は、ゲームシステムの構成を示す外観図であり、図4はそのブロック図である。図3、図4に示すように、ゲームシステムは、ゲーム機本体100、DVD-ROM300、外部メモ리카ード400、コントローラ200、スピーカ600およびTVモニタ500を備える。DVD-ROM300および外部メモ리카ード400は、ゲーム機本体100に着脱自在に装着される。コントローラ200は、通信ケーブルを介して、ゲーム機本体100に設けられた複数（図3では4つ）のコントローラポート用コネクタのいずれかに接続される。TVモニタ500およびスピーカ600は、AVケーブル等によって接続される。なお、ゲーム機本体100とコントローラ200との通信は無線通信であってもよい。以下、図4を参照しながら、ゲームシステムの各部についてより詳細に説明する。

#### 【0021】

DVD-ROM300は、ゲームプログラムや後述するシャドウボリューム生成プログラムやオブジェクトデータ等を固定的に記憶している。プレイヤーがゲームをプレイするとき、DVD-ROM300はゲーム機本体100に装着される。なお、プログラム等を記憶する手段として、DVD-ROM300の代わりに例えばCD-ROM、MO、メモ리카ード、ROMカートリッジ等の外部記憶媒体を用いてもよい。

#### 【0022】

外部メモ리카ード400は、例えばフラッシュメモリ等の書き換え可能な記憶媒体によって構成され、例えばゲームのセーブデータ等のデータを記録する。

#### 【0023】

ゲーム機本体１００は、ＤＶＤ－ＲＯＭ３００に記録されているプログラムを読み出し、読み出したプログラムに応じた処理を行う。

【００２４】

コントローラ２００は、プレイヤがゲーム操作に関する入力を行うための入力装置であり、複数の操作スイッチを有する。コントローラ２００は、プレイヤによる操作スイッチの押圧等に応じて操作データをゲーム機本体１００に出力する。

【００２５】

ＴＶモニタ５００は、ゲーム機本体１００から出力された画像データを画面に表示する。なお、スピーカ６００は、典型的にはＴＶモニタ５００に内蔵されており、ゲーム機本体１００から出力されたゲーム中の音声出力する。

【００２６】

次に、ゲーム機本体１００の構成について説明する。図４において、ゲーム機本体１００には、ＣＰＵ１０およびそれに接続されるメモリコントローラ２０が設けられる。さらにゲーム機本体１００において、メモリコントローラ２０は、ＧＰＵ（グラフィックスプロセッシングユニット）１１と、メインメモリ１７と、ＤＳＰ１８と、各種インターフェース（Ｉ／Ｆ）２１～２４、２６とに接続される。メモリコントローラ２０は、これら各構成要素間のデータ転送を制御する。

【００２７】

ゲーム開始の際、まず、ＤＶＤドライブ２５は、ゲーム機本体１００に装着されたＤＶＤ－ＲＯＭ３００を駆動する。ＤＶＤ－ＲＯＭ３００に記憶されているゲームプログラムは、ＤＶＤディスクＩ／Ｆ２６およびメモリコントローラ２０を介して、メインメモリ１７に読み込まれる。このメインメモリ１７上のプログラムをＣＰＵ１０が実行することによってゲームが開始される。ゲーム開始後、プレイヤは、操作スイッチを用いてコントローラ２００に対してゲーム操作等の入力を行う。プレイヤによる入力に従い、コントローラ２００は、操作データをゲーム機本体１００に出力する。コントローラ２００から出力される操作データは、コントローラＩ／Ｆ２１およびメモリコントローラ２０を介してＣＰＵ１０

に入力される。CPU 10は、入力された操作データに応じてゲーム処理を行う。ゲーム処理における画像データ生成等に際して、GPU 11やDSP 18が用いられる。また、サブメモリ 19は、DSP 18が所定の処理を行う際に用いられる。

#### 【0028】

GPU 11は、ジオメトリユニット 12およびレンダリングユニット 13を含み、画像処理専用のメモリに接続されている。この画像処理専用メモリは、例えばカラーバッファ 14やZバッファ 15やステンシルバッファ 16として利用される。ジオメトリユニット 12は、仮想三次元空間であるゲーム空間に置かれた物体や図形に関する立体モデル（例えばポリゴンで構成されるオブジェクト）の座標についての演算処理を行うものであり、例えば立体モデルの回転・拡大縮小・変形や、ワールド座標系の座標から視点座標系やスクリーン座標系の座標への変換を行うものである。レンダリングユニット 13は、所定のテクスチャに基づいて、スクリーン座標に投影された立体モデルについて各ピクセルごとのカラーデータ（RGBデータ）をカラーバッファ 14に書き込むことによって、ゲーム画像を生成するためのものである。また、カラーバッファ 14は、レンダリングユニット 13によって生成されたゲーム画像データ（RGBデータ）を保持するために確保されたメモリ領域である。Zバッファ 15は、3次元の視点座標から2次元のスクリーン座標に変換する際に失われる視点からの奥行情報を保持するために確保されたメモリ領域である。ステンシルバッファ 16は、シャドウボリューム技法によって影領域を判定するときに利用されるメモリ領域である。GPU 11は、これらバッファを用いてTVモニタ 500に表示すべき画像データを生成し、メモリコントローラ 20およびビデオI/F 22を介して画像データをTVモニタ 500に適宜出力する。なお、ゲームプログラム実行時にCPU 10において生成される音声データは、メモリコントローラ 20からオーディオI/F 24を介してスピーカ 600に出力される。なお本実施形態では、画像処理専用のメモリを別途設けたハードウェア構成としたが、これに限らず例えばメインメモリ 17の一部を画像処理用のメモリとして利用する方式（UMA：Unified Memory Architecture）を使うようにしてもよい。

#### 【0029】

図5に、DVD-ROM300のメモリマップを示す。DVD-ROM300には、ゲームプログラムや、シャドウボリューム生成プログラムや、オブジェクトデータなどが格納されている。なお、シャドウボリューム生成プログラムは、ゲームプログラムに含まれていても構わない。オブジェクトデータには、影を落とすオブジェクトやその他のオブジェクトのデータが含まれる。

#### 【0030】

以下、図6および図7のフローチャートを参照して本ゲームシステムの影描画に関する動作について説明する。

#### 【0031】

図6において、影描画処理が開始すると、CPU10は、まずカラーバッファ14およびZバッファ15をクリアする(S10)。そして前述のシャドウボリューム生成プログラムに基づいて、シャドウボリューム生成処理を実行する(S20)。以下、このシャドウボリューム生成処理の詳細について図7を参照して説明する。

#### 【0032】

図7において、シャドウボリューム生成処理が開始すると、GPU11は、仮想空間に配置された光源を視点として、影を落とすオブジェクトだけをレンダリングする(S201)。つまり、図8(a)に示すように、板1の上に浮かんでいる影を落とすオブジェクト2に対して真上から光があたっている場合、この影を落とすオブジェクト2を真上から見たときのシーンをレンダリングする。その結果、カラーバッファ14には図8(b)に示すような2次元の画像データが書き込まれ、一方、Zバッファ15には、その2次元の画像データの各ピクセルに対応したZ値(視点からオブジェクトまでの距離に応じた値)が書き込まれる。なお、ここでは説明を容易にするためにカラーバッファ14に2次元の画像データが書き込まれるとしたが、実際にはZバッファ15にZ値が書き込まれさえすればよく、カラーバッファ14に画像データを書き込む必要はない。

#### 【0033】

ステップS201のレンダリングが完了すると、CPU10は、つづくステッ

ブ S 2 0 2 ~ S 2 1 4 の処理によって、図 8 ( c ) に示す平面状オブジェクト 3 (図 1 ( b ) に示した柔らかな布に相当) から図 8 ( d ) に示すメッシュモデル 4 を生成する処理を実行する。本実施形態ではこのメッシュモデル 4 がシャドウボリュームとして用いられる。

【0 0 3 4】

平面状オブジェクト 3 は複数のポリゴンによって構成され、これらのポリゴンの頂点でもある複数の頂点を有している。そして各頂点の位置は、X座標とZ座標の組み合わせにより規定されている。本実施形態では、平面状オブジェクト 3 の各頂点の座標がそれぞれ下記のように規定されているものとする。

$$(0, 0) \quad , \quad (1, 0) \quad , \quad (2, 0) \quad , \quad \cdots \quad , \quad (x_m, 0)$$
$$(0, 1) \quad , \quad (1, 1) \quad , \quad (2, 1) \quad , \quad \cdots , \quad (x_m, 1)$$
$$(0, 2) \quad , \quad (1, 2) \quad , \quad (2, 2) \quad , \quad \cdots , \quad (x_m, 2)$$
$$(0, 3) \quad , \quad (1, 3) \quad , \quad (2, 3) \quad , \quad \cdots \quad , \quad (x_m, 3)$$
$$(0, z_m), (1, z_m), (2, z_m), \dots, (x_m, z_m)$$

なお以下の説明では、メッシュモデル4のある頂点Pについて、その元となった平面状オブジェクト3の頂点の座標（ $x$ ， $z$ ）を、メッシュモデル4の頂点Pの基本座標と称する。そして、特に基本座標のうちのX座標およびZ座標を、それぞれ基本X座標および基本Z座標と称する。

【0035】

以下、ステップS202以降の処理について具体的に説明する。

なお、図 7 に示すフローチャートにおいて、

- ・  $x$  および  $z$  は、それぞれ基本 X 座標および基本 Z 座標を示す変数であり、
- ・  $x_m$  および  $z_m$  は、それぞれ基本 X 座標の上限値および基本 Z 座標の上限値を示す変数であり、
- ・  $Z_{a d r s}$  は、Z バッファ 15 のアドレスを示す変数であり、
- ・  $Z_{v a l u e}$  は、Z バッファ 15 の値を示す変数であり、
- ・  $P_{a d r s}$  は、メッシュモデル 4 の頂点座標を格納するメインメモリ 17 のア

ドレスを示す変数であり、

- ・ `p x`、`p y`および`p z`は、メッシュモデル4のX座標、Y座標およびZ座標をそれぞれ示す変数である。

#### 【0036】

また、図7に示すフローチャートにおいて、

- ・ `cal_Zadr s()`は、基本座標から、対応するZバッファ15のアドレスを求める関数であり、

- ・ `cal_Padr s()`は、基本座標から、対応するメッシュモデル4の頂点データ格納アドレスを求める関数であり、

- ・ `read_Zbuf()`は、Zバッファアドレスから、そのZ値を読み込む関数であり、

- ・ `cal_height()`は、Z値から適切な高さの値を計算する関数であり、

- ・ `scale_mesh()`は、メッシュモデル4の各頂点がZ値に応じて（つまりZ値に対して比例的に変化するように、またはZ値に応じてべき乗的に変化するように、または予め用意された変換テーブルに基づいてZ値に応じて所定の増加率で変化するように）放射状に移動するように（つまり光源から遠い頂点ほど大きく移動するように）、基本X座標または基本Z座標を修正する関数であり、

- ・ `store_pos()`は、X座標、Y座標およびZ座標をメッシュモデル4の頂点データ格納領域に格納する関数である。

#### 【0037】

ステップS202において、CPU10は、基本Z座標を初期化し、続くステップS203においてさらに基本X座標を初期化する。つまり、メッシュモデル4の形状を規定する複数の頂点のうち、基本座標が(0, 0)である頂点が最初の処理対象頂点として選ばれることになる。

#### 【0038】

ステップS204で、CPU10は、処理対象頂点の基本座標(x, y)から、この基本座標に対応するZバッファ15のアドレスおよび頂点データの格納ア

ドレスを求める。そして、このZバッファ15のアドレスのZ値を読み込む（S205）。つまり、カラーバッファ14に格納されている図8（b）に示す2次元画像を構成するピクセルのうち、処理対象頂点に対応するピクセルのZ値が、Zバッファ15より読み出されることになる。CPU10は、こうして読み出したZ値から、処理対象頂点のY座標（つまり平面状オブジェクト3の厚み方向に関する位置）を決定する（S206）。Z値から処理対象頂点のY座標（py）を求めるには、例えば図9に示すように、視点（光源）から遠い側のクリッピングプレーンに位置するZ値（例えば図9では1）と、レンダリングされたZ値（Z value）の差を求め、それをY座標軸の単位系にスケーリングすればよい。

#### 【0039】

ステップS206で処理対象頂点のY座標が決定されると、次にCPU10は、光源が平行光源であるか点光源であるかを判断する（S207）。そして、光源が平行光源であった場合にはステップS208に進み、光源が点光源であった場合にはステップS209に進む。なお、光源として平行光源のみもしくは点光源のみしか用意されていないシステムにおいては、ステップS207が不要であることは言うまでもない。

#### 【0040】

光源が平行光源だった場合には、図8（d）に示すように、処理対象頂点のX座標（px）およびZ座標（pz）として、それぞれ基本X座標（x）および基本Z座標（z）をそのまま利用する（S208）。一方、光源が点光源だった場合には、ステップS205において読み出されたZ値（Z value）に応じて基本Z座標（x）および基本Z座標（z）を修正したものを、処理対象頂点のX座標（px）およびZ座標（pz）として設定する（S209）。このように光源の種類によって処理が変わる理由について図10を参照して説明する。

#### 【0041】

図10（a）に示すように、影を落とすオブジェクトが平行光源によって照らされている場合には、ステップS201のレンダリング処理において、奥行きが変わっても見える大きさが同じである正射影によって影を落とすオブジェクトが

描画され、図10(e)に示すように、点光源によって照らされている場合には、ステップS201のレンダリング処理において、奥に位置するほど小さく見える透視射影によって影を落とすオブジェクトが描画される。その結果、平行光源の場合と点光源の場合とで、カラーバッファ14に格納される2次元画像データの内容が、それぞれ図10(b)および図10(f)に示すように異なることになる。これは、Zバッファ15に格納されるZ値の内容についても同様である。そして、そのようなZ値に基づいてステップS206が実行された結果、平面状オブジェクトの厚み方向に関する位置が、平行光源の場合には図10(c)に示すように決定され、点光源の場合には図10(g)に示すように決定される。ここで、平行光源の場合には、図10(c)のメッシュモデルをそのまま図10(d)に示すようにシャドウボリュームとして利用することが可能であるが、点光源の場合には、図10(g)のメッシュモデルをそのままシャドウボリュームとして利用することができない。つまり点光源の場合には、図10(h)に示すように、メッシュモデルの各頂点が、平面状オブジェクト3の面方向に関してZ値に応じて放射状に移動するように（つまり光源から遠い頂点ほど大きく移動するように）、基本X座標または基本Z座標を修正する。なお、放射状に移動するときの中心軸は、ステップS201において光源を視点として影を落とすオブジェクトをレンダリングしたときに生成された2次元画像の中心点に対応する。このようにして生成されたメッシュモデルは、図10(i)に示すようにシャドウボリュームとして利用可能となる。

#### 【0042】

ステップS208またはステップS209において処理対象頂点のX座標およびZ座標が決定されると、CPU10は、処理対象頂点の座標をステップS204で求められたアドレスに格納する(S210)。その後、基本X座標(x)をインクリメントし(S211)、基本X座標が最大値(x<sub>m</sub>)を超えるまで処理対象頂点を順次変更しながらステップS204～S212の処理を繰り返す(ステップS212でNo)。そして、基本X座標が最大値を超えた場合には(ステップS212でYes)、基本Z座標(z)をインクリメントし(S213)、続いて基本Z座標が最大値(z<sub>m</sub>)を超えたかどうかを判断する(S214)。



判断の結果、Z座標値が最大値を超えていなかった場合には（ステップS214でNo）ステップS203に戻り、Z座標値が最大値を超えていた場合には（これはメッシュモデル4の全頂点の頂点データがメインメモリ17に格納されたことを意味する）、シャドウボリューム生成処理を終了し、図6のフローチャートに処理が戻る。

#### 【0043】

図6において、シャドウボリューム生成処理が完了すると、GPU11は、Zバッファ15と、必要に応じてカラーバッファ14をクリアしてから（S30）、シーンをレンダリングする（S40）。ここでは通常のレンダリング処理が行われ、例えば図8（a）に示す板1および影を落とすオブジェクト2がレンダリングされる。

#### 【0044】

シーンのレンダリングが完了後、GPU11は、ステップS20で生成したメッシュモデル4をシャドウボリュームとして利用して、シャドウボリューム技法による影描画処理を実行する。具体的には、図11（a）に示す仮想空間において、図11（b）に示すメッシュモデル4が図11（c）のように仮想的に配置され、ステンシルバッファ16を用いたシャドウボリューム技法によって図11（d）に示す影の領域が抽出される。

#### 【0045】

なお、本実施形態では、影を落とすオブジェクト2が上方から光に照らされている場合について説明したが、影を落とすオブジェクト2が斜め上方から光に照らされている場合についても同様に本発明を適用することができる。例えば、図12（a）のように、球状の影を落とすオブジェクト2に対して斜め上方から光があたっている場合、この影を落とすオブジェクト2を光源を視点としてレンダリングして得られる2次元画像は図12（b）のようになる。そして、レンダリングの結果としてZバッファ15に書き込まれているZ値に基づいて、平面状オブジェクト3の各頂点の厚み方向に関する位置が決定され、最終的に図12（c）に示すメッシュモデル4が得られる。このメッシュモデル4をシャドウボリュームとして利用して影描画を行うときには、まず図13（a）に示す仮想空間に

において、図13(b)に示すメッシュモデル4が図13(c)のように仮想的に配置される。つまり、光線方向と、メッシュモデル4の厚み方向(つまりY軸方向)とが平行になるようにメッシュモデル4を配置する。その後、ステンシルバッファ16を用いたシャドウボリューム技法によって図13(d)に示す影の領域が抽出される。

#### 【0046】

また、本実施形態では、平面状オブジェクト3に基づいてシャドウボリュームを生成するとしたが、この平面状オブジェクト3が、例えば図14(a)に示すように立体オブジェクト5の一面を構成するものであっても構わない。この場合にも、本実施形態と同様に、Zバッファ15に書き込まれているZ値に基づいて平面状オブジェクト3の各頂点の厚み方向に関する位置を決定することによって、図14(b)に示すようなシャドウボリュームが得られ、このシャドウボリュームを利用して影描画を行うことができる。

#### 【0047】

以上のように、本実施形態によれば、影を落とすオブジェクトの形状の複雑さに関係なく、常に同じ処理によってどんな形状のオブジェクトのシャドウボリュームも生成することができる。したがって、例外処理が発生せず、プログラムが少量化されると同時に安定した処理が可能となる。さらに、本実施形態で行われる処理は、グラフィックプロセッサの機能を利用して実行することが可能であるため、グラフィックプロセッサに処理を任せることによって処理速度が大幅に向上する。

#### 【0048】

なお本実施形態ではゲームプログラムはDVD-ROM300を介してゲーム機本体100に供給されたとしたが、これに限らず、DVD-ROM300以外の他のコンピュータ読み取り可能な記録媒体(例えばCD-ROM、MO、メモリカード、ROMカートリッジ等)に格納されてゲーム機本体100に供給されても構わない。また、予めゲーム機本体100に組み込まれていても構わない。また、インターネットなどの通信回線を通じてゲーム機本体100に供給されても構わない。

【００４９】

【発明の効果】

以上のように、本発明によれば、例外的な処理を必要とせず、常に一定の簡単な処理でシャドウボリュームを作成することができる。

【図面の簡単な説明】

【図１】

本発明の原理を説明するための図である。

【図２】

シャドウボリュームと影の関係を示す図である。

【図３】

本発明の一実施形態に係るゲームシステムの外觀図である。

【図４】

ゲーム機本体１００の構成を示すブロック図である。

【図５】

DVD-ROM３００のメモリマップである。

【図６】

影描画処理の流れを示すフローチャートである。

【図７】

シャドウボリューム生成処理の流れを示すフローチャートである。

【図８】

シャドウボリューム生成処理の具体例を示す図である。

【図９】

Z値からメッシュモデルの各頂点のY座標を決定する具体例を示す図である。

【図１０】

光源が平行光源である場合と点光源である場合との処理の相違点を説明するための図である。

【図１１】

シャドウボリュームを用いた影描画処理の具体例を示す図である。

【図１２】

影を落とすオブジェクトの斜め上方から光があたっている場合のシャドウボリューム生成処理の具体例を示す図である。

【図 1 3】

影を落とすオブジェクトの斜め上方から光があたっている場合のシャドウボリュームの配置方法を説明するための図である。

【図 1 4】

平面状オブジェクトが立体オブジェクトの一部である場合のシャドウボリューム生成処理の具体例を示す図である。

【図 1 5】

従来のシャドウボリューム技法における輪郭線の抽出処理を示す図である。

【図 1 6】

従来のシャドウボリューム技法において抽出される輪郭線の形を示す図である。

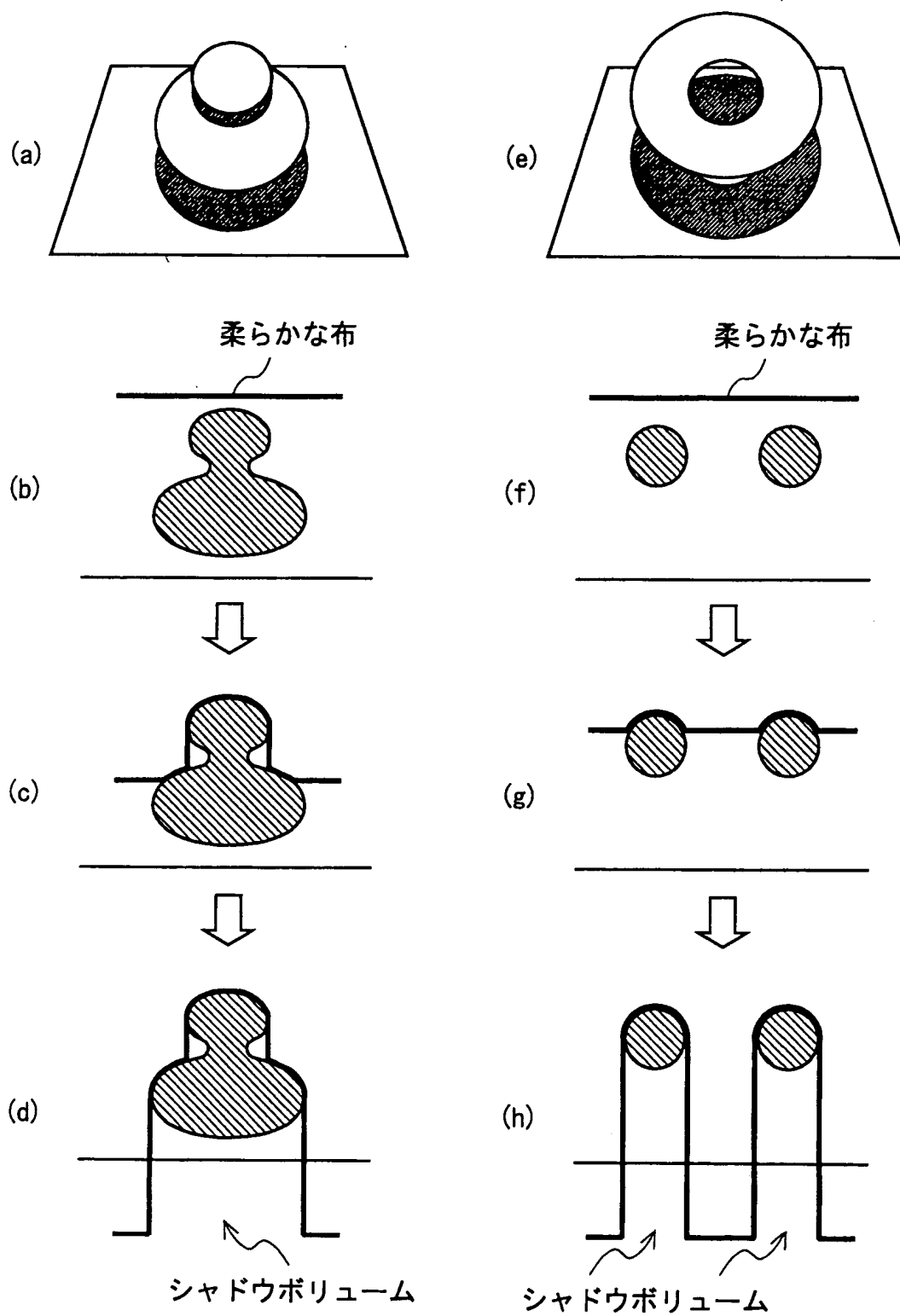
【符号の説明】

- 1 板
- 2 影を落とすオブジェクト
- 3 平面状オブジェクト
- 4 メッシュモデル
- 5 立体オブジェクト
- 10 CPU
- 11 GPU
- 12 ジオメトリユニット
- 13 レンダリングユニット
- 14 カラーバッファ
- 15 Zバッファ
- 16 ステンシルバッファ
- 17 メインメモリ
- 18 DSP
- 19 サブメモリ

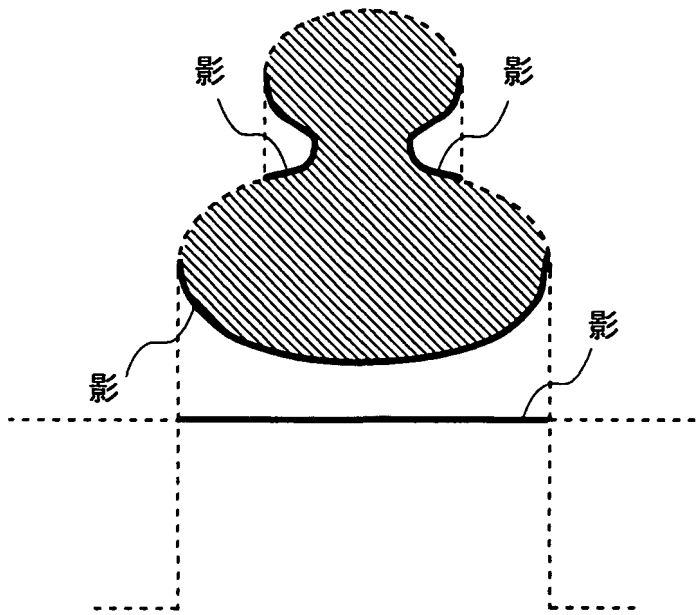
20 メモリコントローラ  
21 コントローラ I/F  
22 ビデオ I/F  
23 外部メモリ I/F  
24 オーディオ I/F  
25 DVDドライブ  
26 DVDディスク I/F  
100 ゲーム機本体  
200 コントローラ  
300 DVD-ROM  
400 外部メモリカード  
500 TVモニター  
600 スピーカ

【書類名】 図面

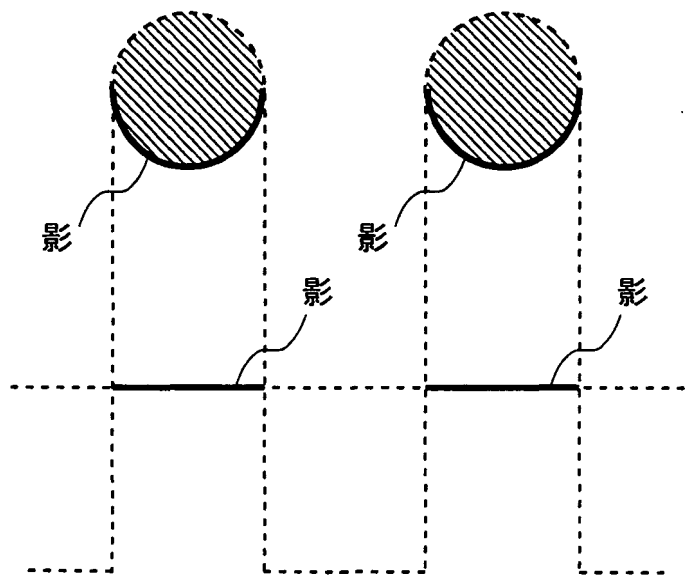
【図 1】



【図 2】

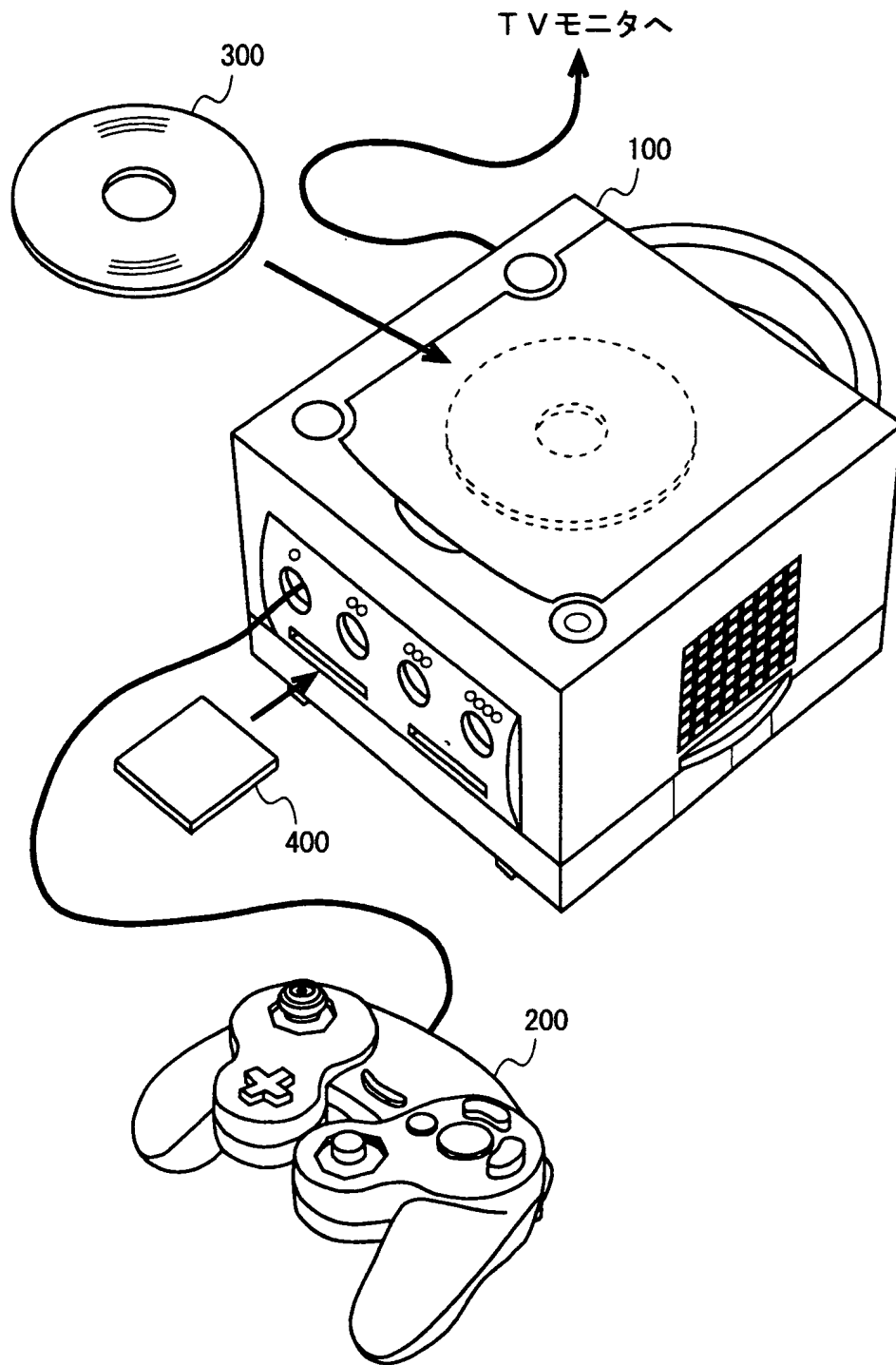


(a)



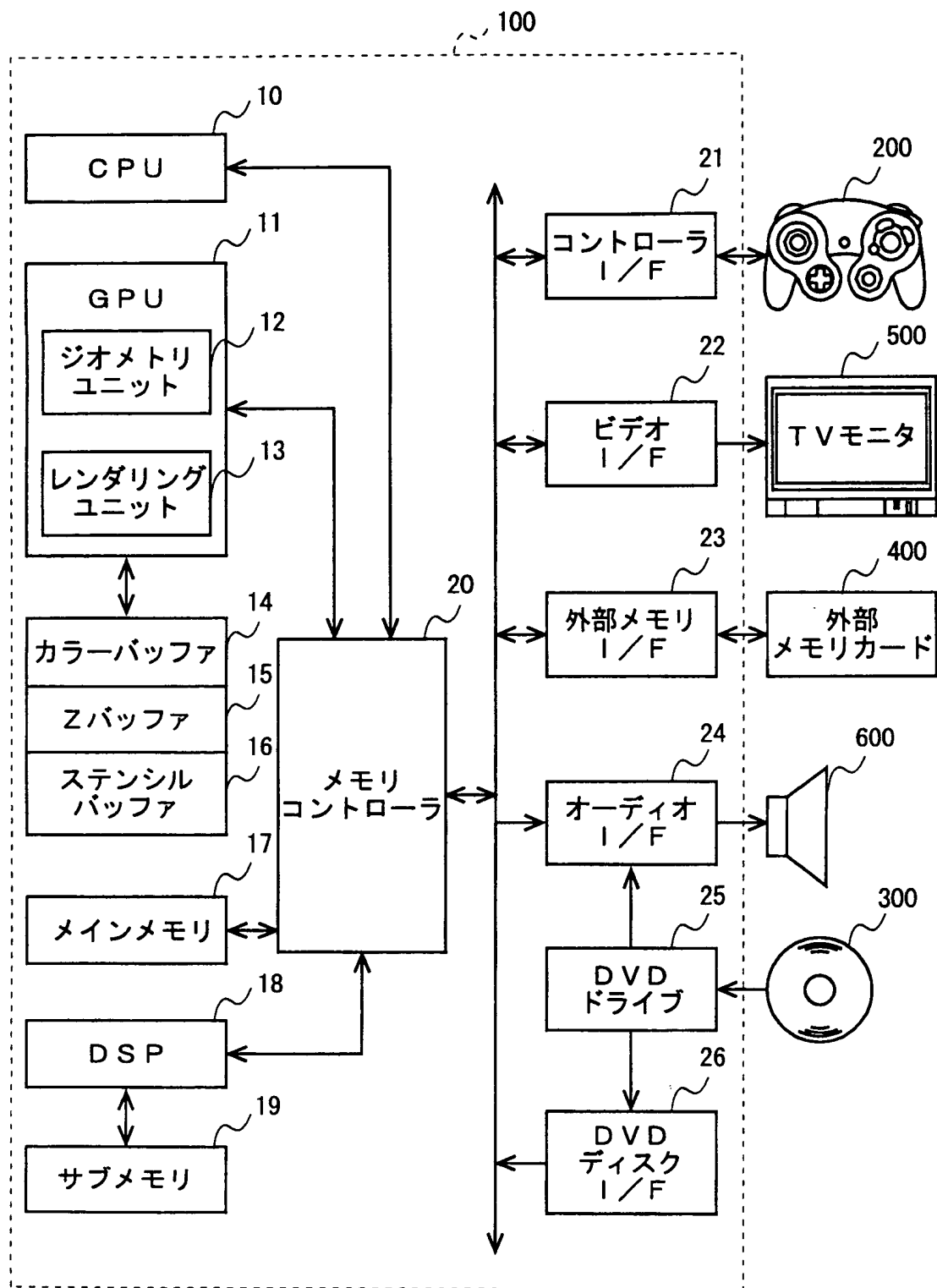
(b)

【図 3】

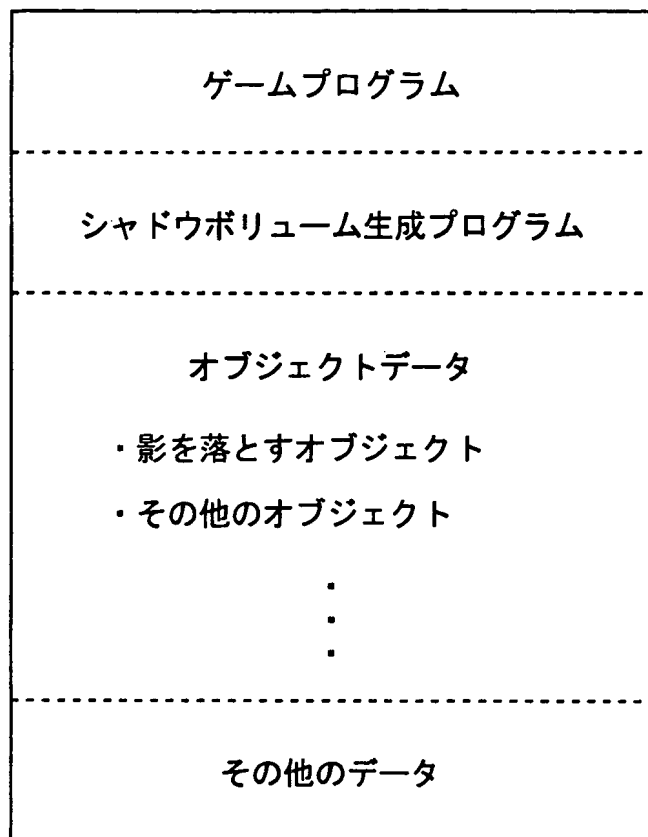




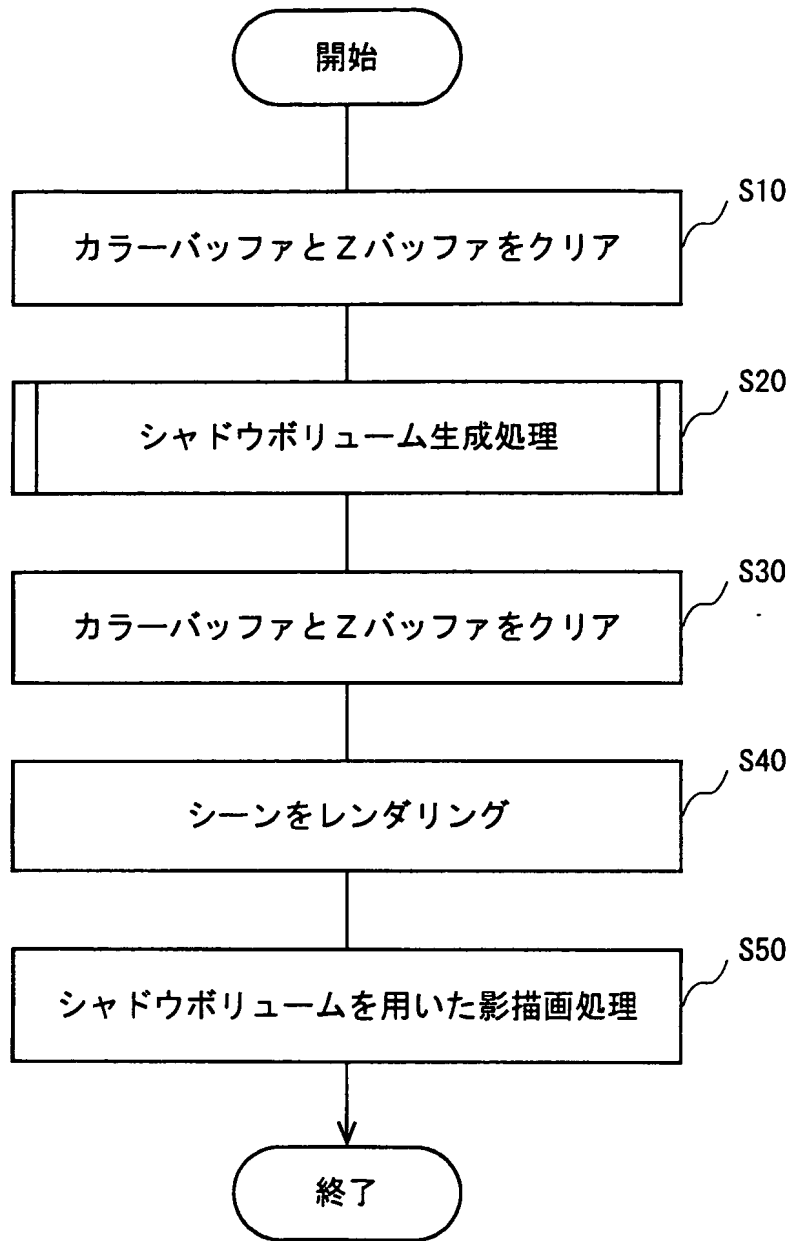
【図 4】



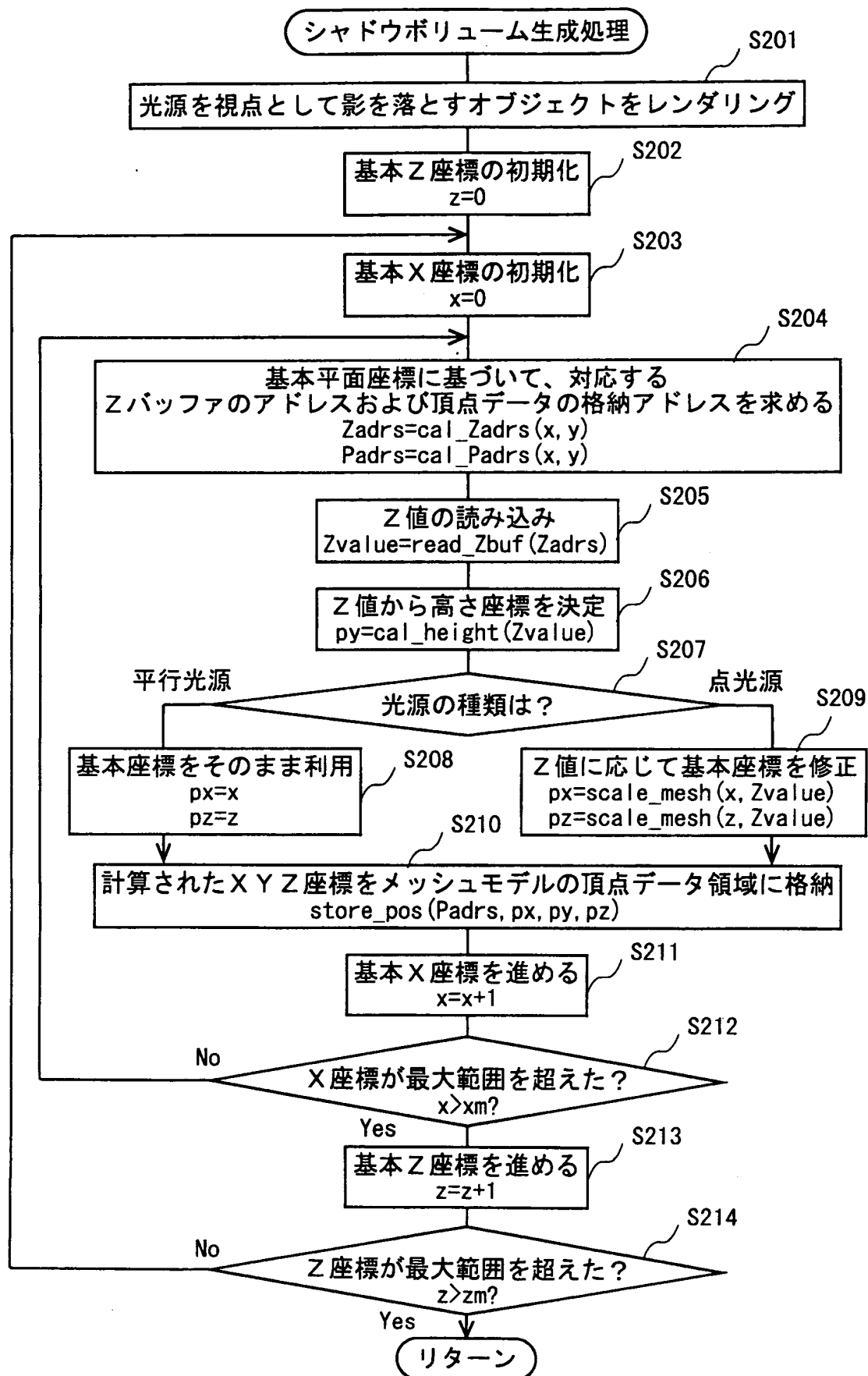
【図5】



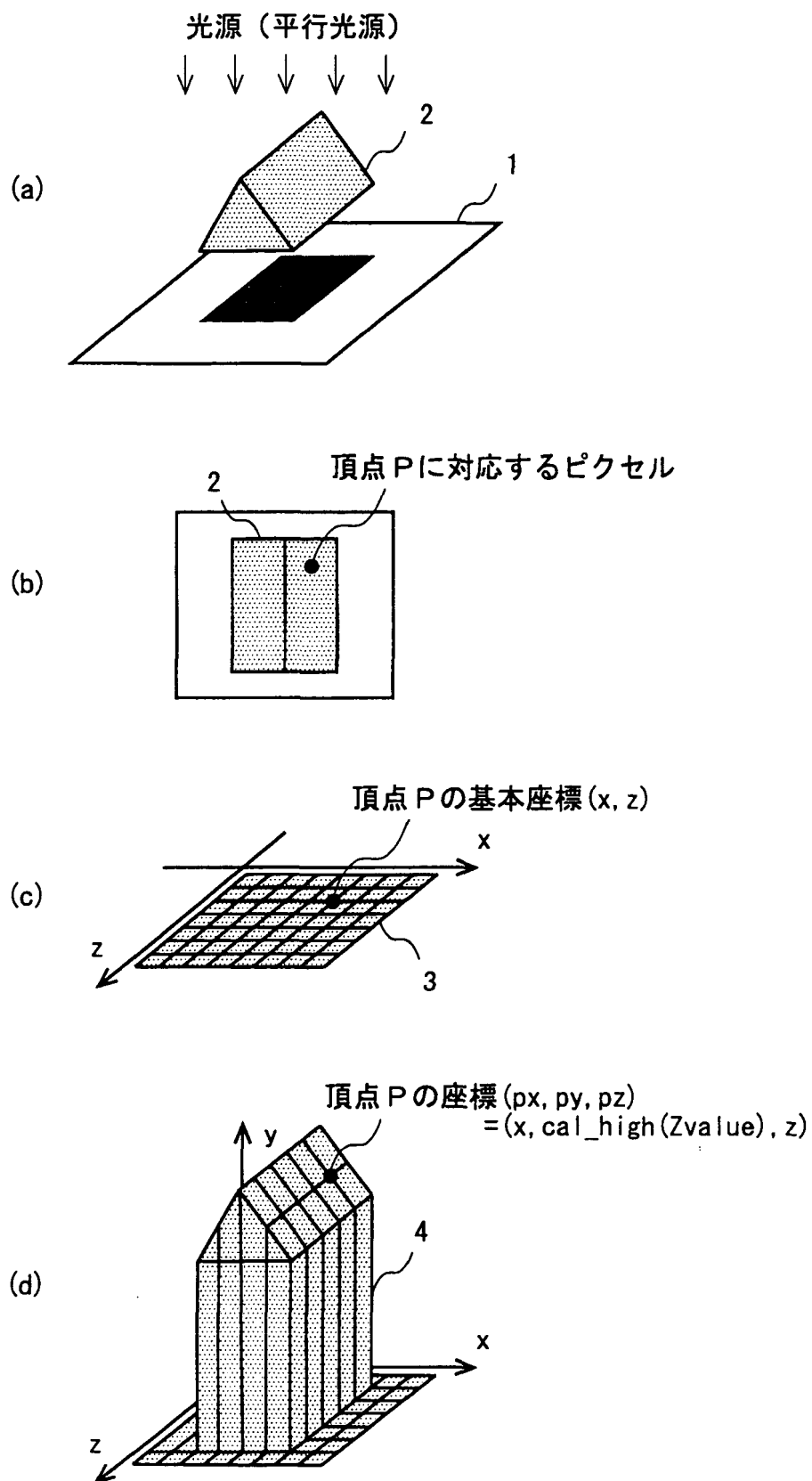
【図 6】



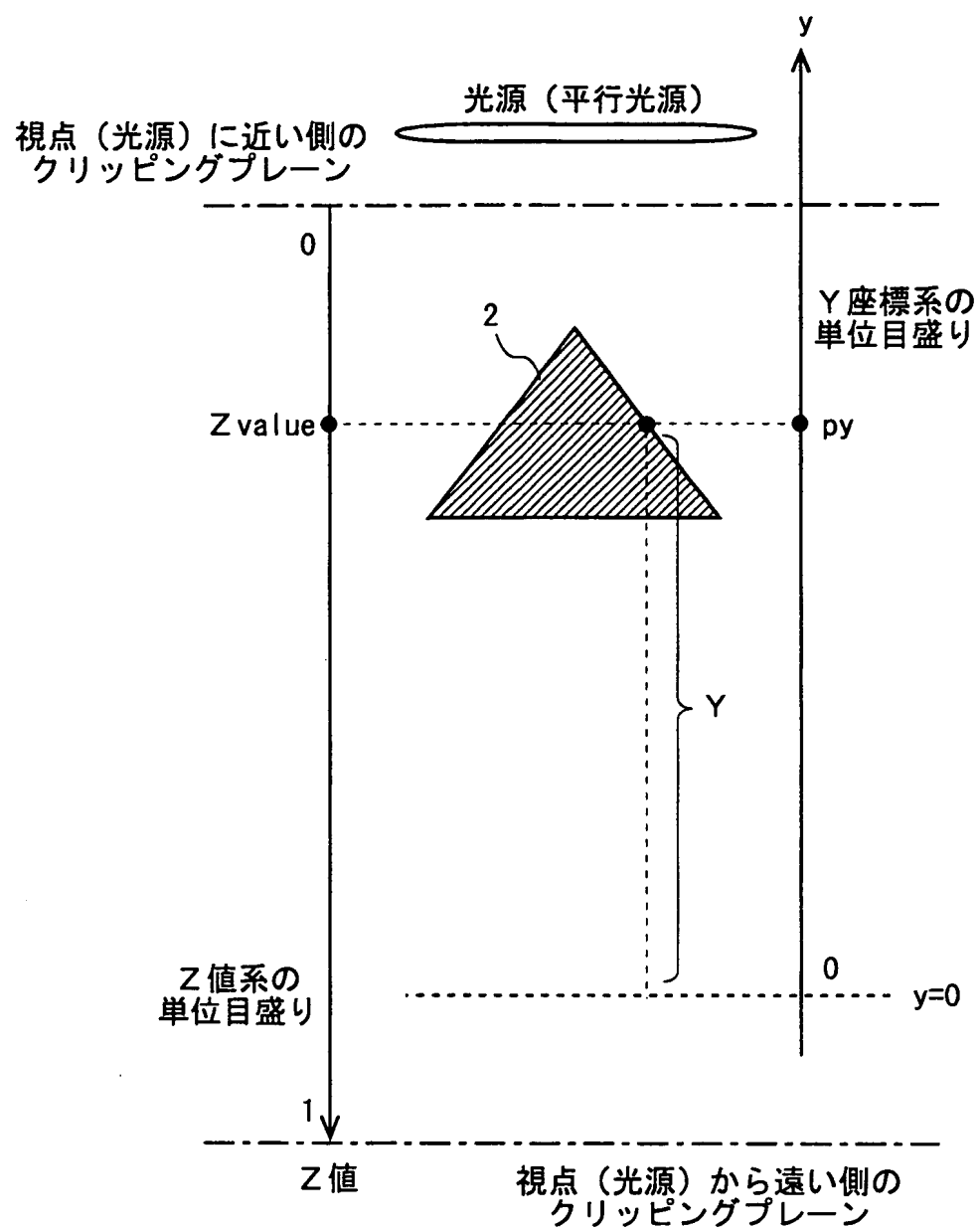
【図 7】



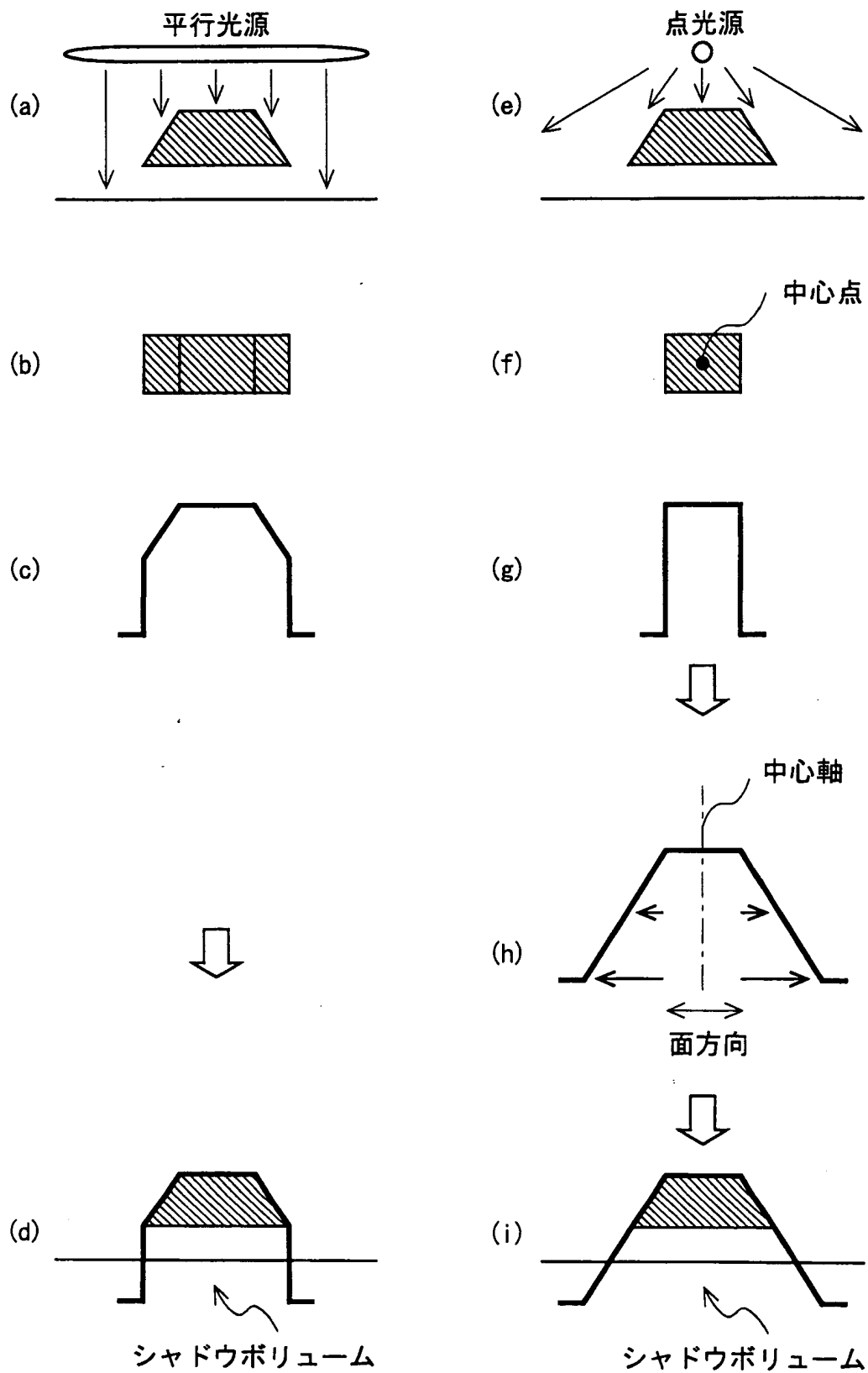
【図 8】



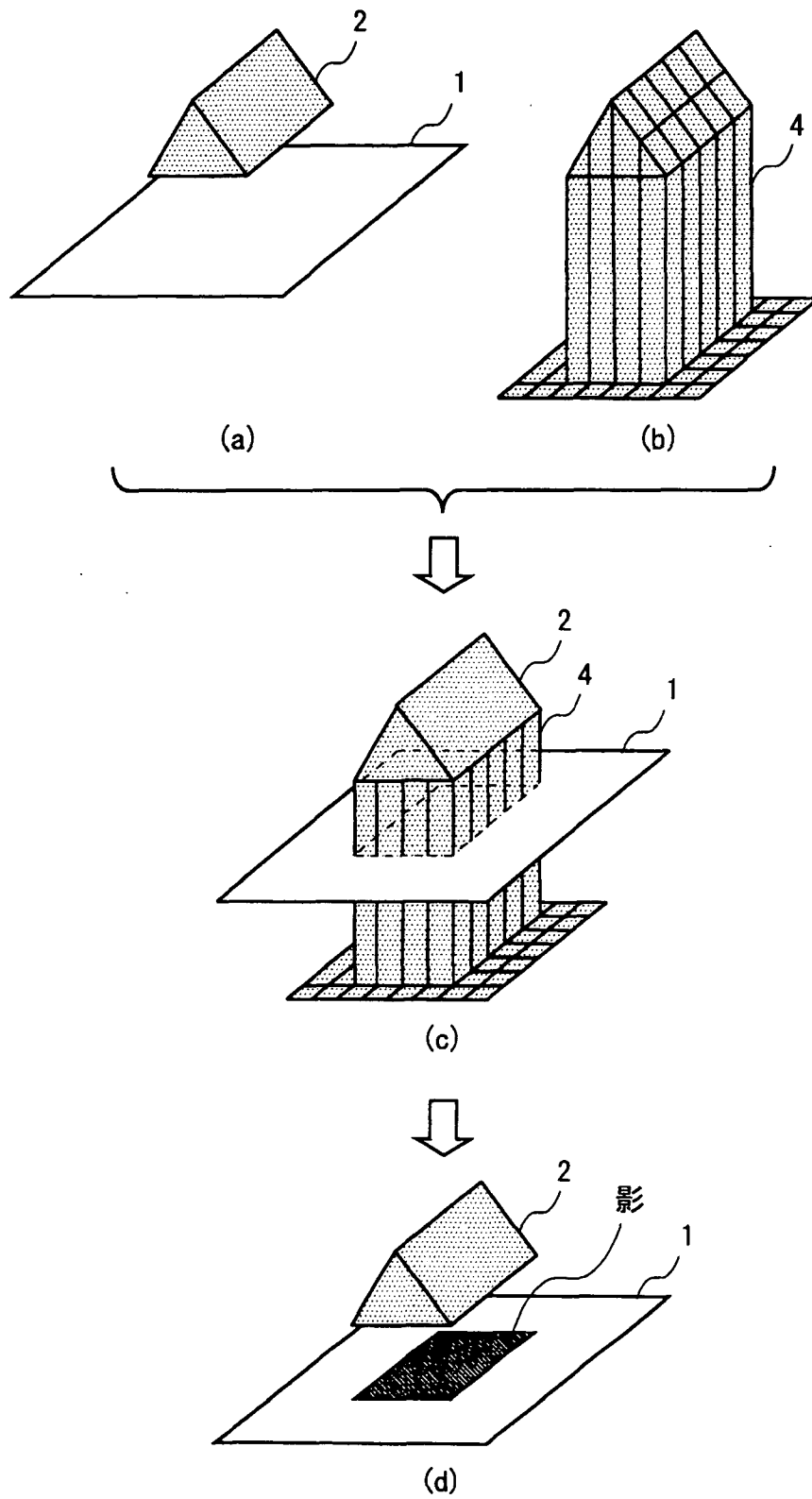
【図 9】



【図 10】

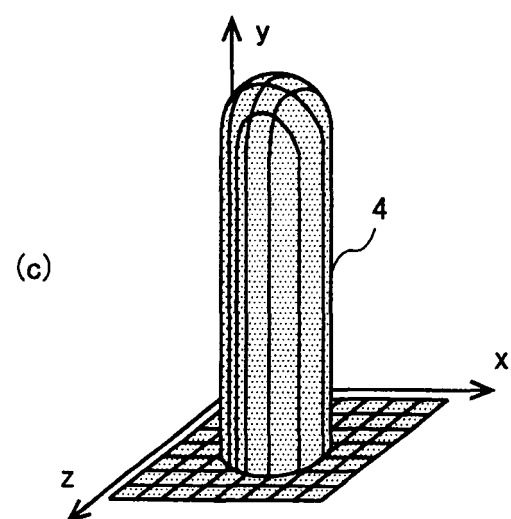
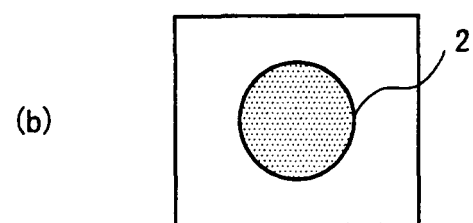
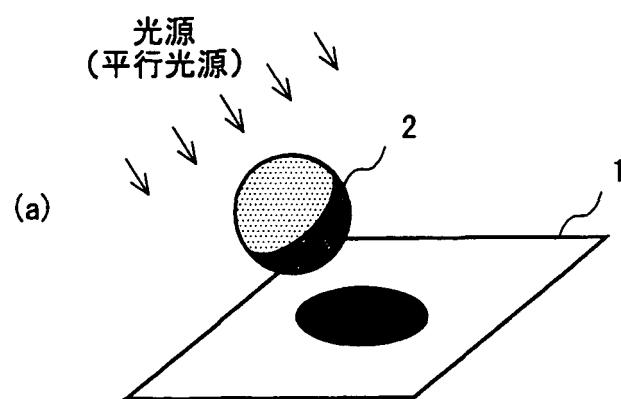


【図 1 1】

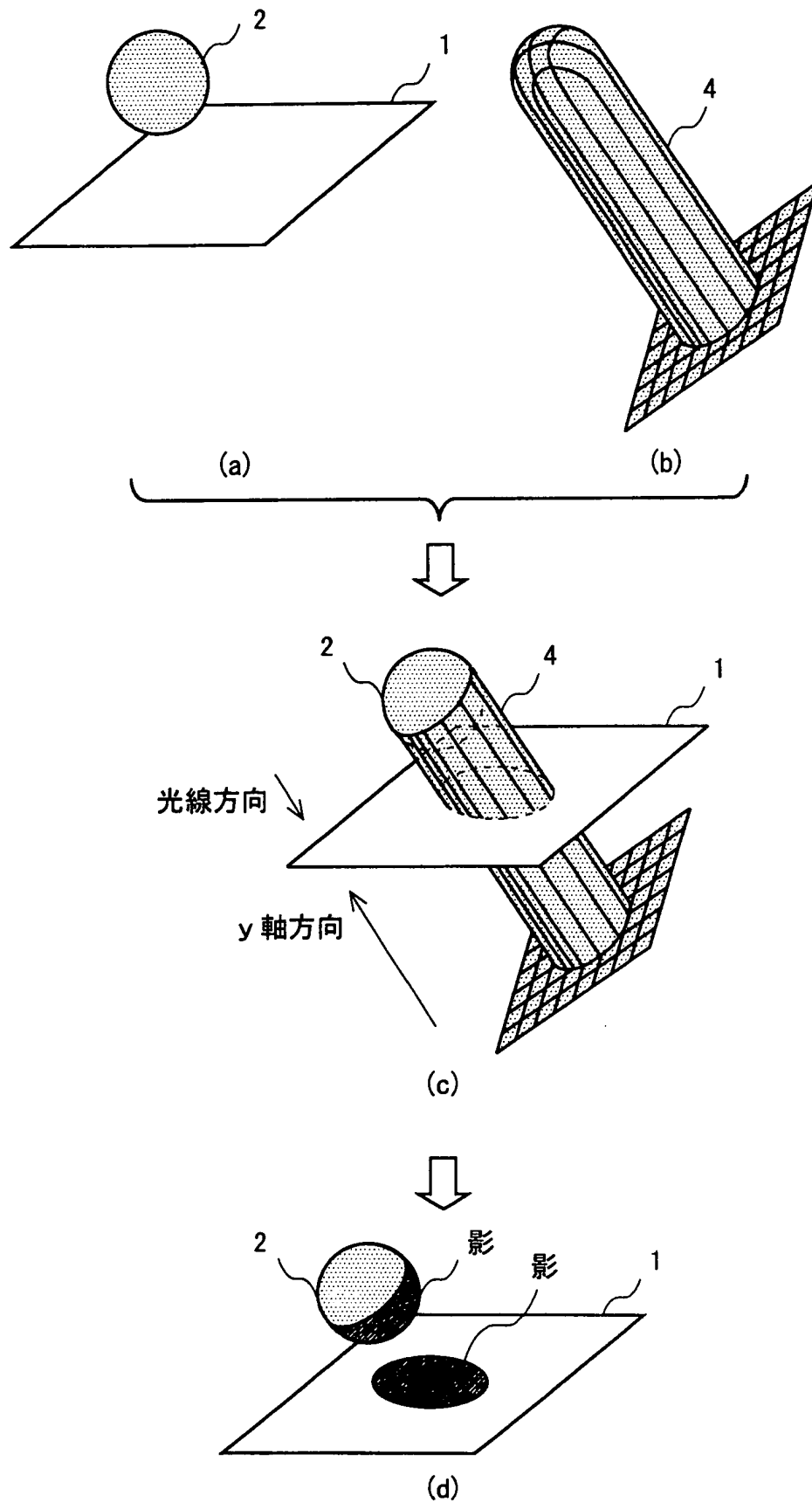




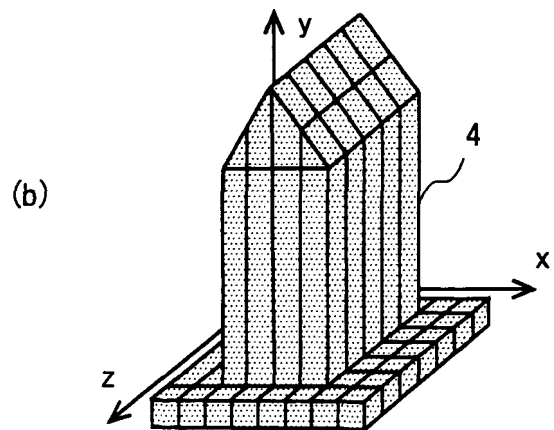
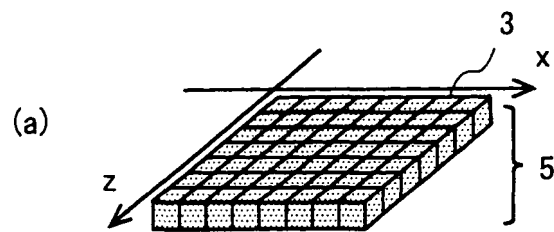
【図 1 2】



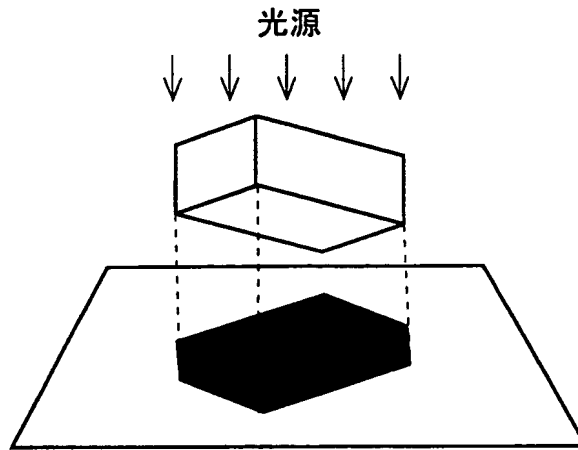
【図13】



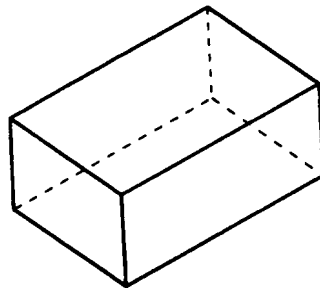
【図 1 4】



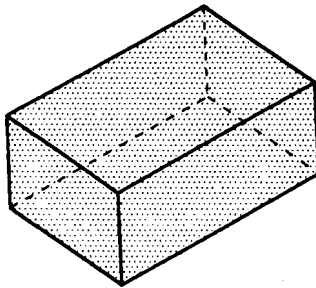
【図 1 5】



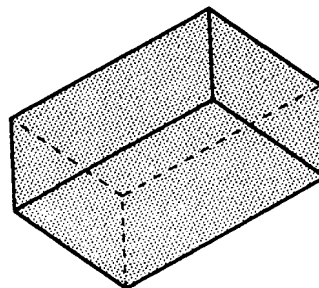
(a) 光源と影オブジェクトの位置関係



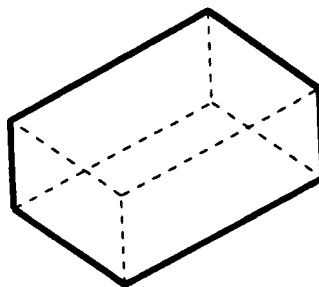
(b) 光源から見た影オブジェクト



(c) 光源に対して表向きのポリゴン

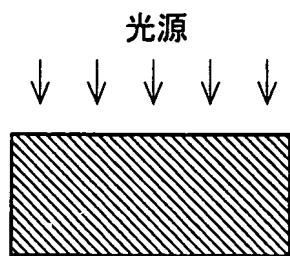


(d) 光源に対して裏向きのポリゴン

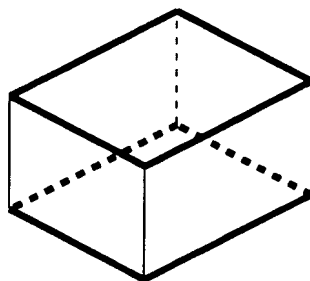


(e) 表面と裏面の両方に接するエッジ

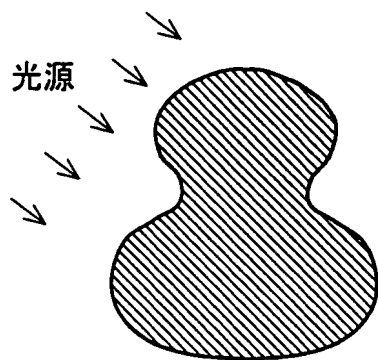
【图 1 6】



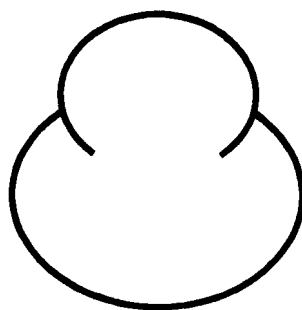
(a)



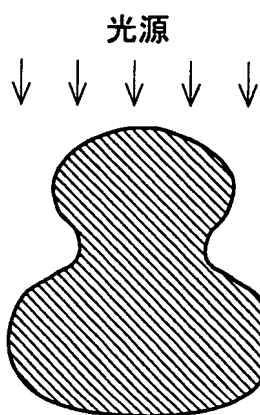
(b)



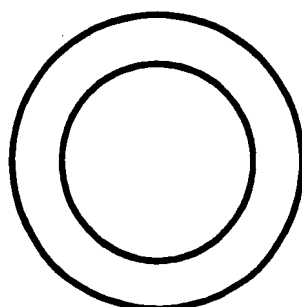
(c)



(d)



(e)



(f)

【書類名】 要約書

【要約】

【課題】 例外的な処理を必要とせず、常に一定の簡単な処理でシャドウボリュームを作成すること。

【解決手段】 仮想空間に配置された光源を視点として、影を落とすオブジェクト2だけがレンダリングされ、その結果、Zバッファには、光源方向からオブジェクトをとらえた2次元画像の各ピクセルに対応したZ値が書き込まれる。レンダリングが完了すると、平面状オブジェクト3からメッシュモデル4が生成される。つまり、平面状オブジェクト3の各頂点に対応するピクセルのZ値がZバッファより読み出され、このZ値から各頂点のY座標が決定される。こうして生成されたメッシュモデル4を利用して、シャドウボリューム技法による影描画処理が実行される。

【選択図】 図8

## 認定・付加情報

特許出願の番号	特願 2 0 0 3 - 0 7 2 5 5 2
受付番号	5 0 3 0 0 4 3 4 9 3 5
書類名	特許願
担当官	第二担当上席 0 0 9 1
作成日	平成 1 5 年 3 月 1 8 日

### <認定情報・付加情報>

【提出日】	平成15年 3月17日
-------	-------------

次頁無

出 願 人 履 歴 情 報

識別番号 [000233778]

1. 変更年月日 2000年11月27日

[変更理由] 住所変更

住 所 京都府京都市南区上鳥羽鉾立町11番地1

氏 名 任天堂株式会社